

NetSim Multi-Parameter Sweep Program

Software Recommended:

NetSim v13.0 (32/64 bit), [DOT NET CORE SDK 3.1](#), [Python 3.7.4](#)

Project Download Link:

https://github.com/NetSim-TETCOS/MODELLING_OBSTACLES_IN_LTE_v13.0/archive/refs/heads/main.zip

Introduction:

When users want to *sweep* one or more parameters, they change their values between simulation runs, and compare and analyse the performance metrics from each run. NetSim multi-parameter sweeper enables users to automate the sweep process.

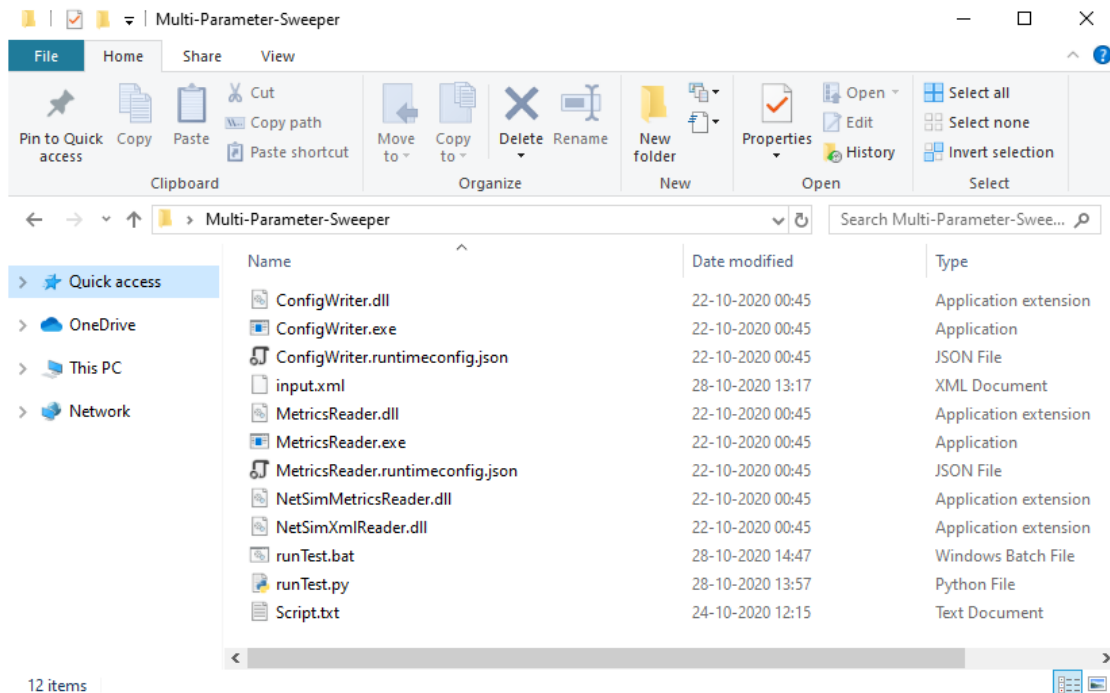
Consider an example, where a user wishes to create and simulate a network scenario for all possible values of one or more parameters in combination and analyse a set of performance metrics across the simulation runs. This is be extremely time consuming to do manually using the NetSim GUI.

The multi-parameter sweep program enables users to automate the sweep process across multiple input parameters, simulate each run, save each result, and compare specific output metrics via a spreadsheet software like MS Excel.

The sweep program runs NetSim via its CLI interface.

File Organization

The project directory consists of several binaries which are responsible for different tasks during a multi-parameter sweep:



1. **input.xml**: This file contains the base NetSim network configuration that is to be simulated. This file is created by copy pasting the Configuration.netsim file that can be obtained by saving a network configuration in NetSim and renaming it to input.xml.

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <TETCOS_NETSIM xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance" ns0:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance">
3   <EXPERIMENT_INFORMATION>...</EXPERIMENT_INFORMATION>
11  <GUI_INFORMATION>...</GUI_INFORMATION>
18  <NETWORK_CONFIGURATION>
19    <DEVICE_CONFIGURATION DEVICE_COUNT="4">
20      <DEVICE DEFAULT_DEVICE_NAME="gNB" DEVICE_ID="1" DEVICE_IMAGE="gNB.png" DEVICE_NAME="gNB">
68      <DEVICE DEFAULT_DEVICE_NAME="EPC" DEVICE_ID="2" DEVICE_IMAGE="EPC.png" DEVICE_NAME="EPC">
131     <DEVICE DEFAULT_DEVICE_NAME="Wired_Node" DEVICE_ID="3" DEVICE_IMAGE="WiredNode.png" DEVICE_NAME="Wired_Node">
168     <DEVICE DEFAULT_DEVICE_NAME="UE" DEVICE_ID="4" DEVICE_IMAGE="UserEquipment.png" DEVICE_NAME="UE">
199    </DEVICE_CONFIGURATION>
200    <CONNECTION>
201      <LINK DEVICE_COUNT="2" LINK_COLOR="" LINK_ID="1" LINK_MODE="FULL DUPLEX" LINK_NAME="LINK_1">
206      <LINK DEVICE_COUNT="2" LINK_COLOR="" LINK_ID="2" LINK_MODE="FULL DUPLEX" LINK_NAME="LINK_2">
212      <LINK DEVICE_COUNT="2" LINK_COLOR="1885ad" LINK_ID="3" LINK_MODE="HALF DUPLEX" LINK_NAME="LINK_3">
217    </CONNECTION>
218    <APPLICATION_CONFIGURATION COUNT="1">
219      <APPLICATION APPLICATION_COLOR="0x9000ffff" APPLICATION_METHOD="UNICAST" APPLICATION_NAME="APPLICATION_1">
223    </APPLICATION_CONFIGURATION>
224  </NETWORK_CONFIGURATION>
225 </TETCOS_NETSIM>

```

The values of parameters which are to be varied during each simulation run needs to be specified as {0}, {1}, {2}, etc. respectively.

For Example, if the X and Y coordinates of a device is to be varied the values can be modified in the input.xml file as shown below:

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <TETCOS_NETSIM xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance" ns0:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance">
3   <EXPERIMENT_INFORMATION>...</EXPERIMENT_INFORMATION>
11  <GUI_INFORMATION>...</GUI_INFORMATION>
18  <NETWORK_CONFIGURATION>
19    <DEVICE_CONFIGURATION DEVICE_COUNT="4">
20      <DEVICE DEFAULT_DEVICE NAME="gNB" DEVICE_ID="1" DEVICE_IMAGE="gNB.png" DEVICE_NAME="gNB" DEVICE_DEFAULT_DEVICE NAME="EPC" DEVICE_ID="2" DEVICE_IMAGE="EPC.png" DEVICE_NAME="EPC"
68      <DEVICE DEFAULT_DEVICE NAME="Wired Node" DEVICE_ID="3" DEVICE_IMAGE="WiredNode.png" DEVICE_NAME="Wired Node"
131      <DEVICE DEFAULT_DEVICE NAME="UE" DEVICE_ID="4" DEVICE_IMAGE="UserEquipment.png" DEVICE_NAME="User Equipment"
168      <POS_3D X_OR_LON="{0}" Y_OR_LAT="{1}" Z="{0}">
169        <MOBILITY MODEL="NO_MOBILITY"/>
170      </POS_3D>
171      <INTERFACE ID="1" INTERFACE_TYPE="LTE_NR">
172        <LAYER TYPE="NETWORK_LAYER">
173          <NETWORK_PROTOCOL NAME="IPV4" SETPROPERTY="TRUE">
174            <PROTOCOL_PROPERTY DEFAULT_GATEWAY="11.2.1.1" IP_ADDRESS="11.2.1.2" SUBNET_MASK="255.255.255.0">
175              <PROTOCOL_PROPERTY_DEFAULT_GATEWAY="11.2.1.1" IP_ADDRESS="11.2.1.2" SUBNET_MASK="255.255.255.0">
176            </PROTOCOL_PROPERTY>
177          </NETWORK_PROTOCOL>
178        </LAYER>
179      </INTERFACE>
180    </DEVICE_CONFIGURATION>
181  </NETWORK_CONFIGURATION>
182</TETCOS_NETSIM>

```

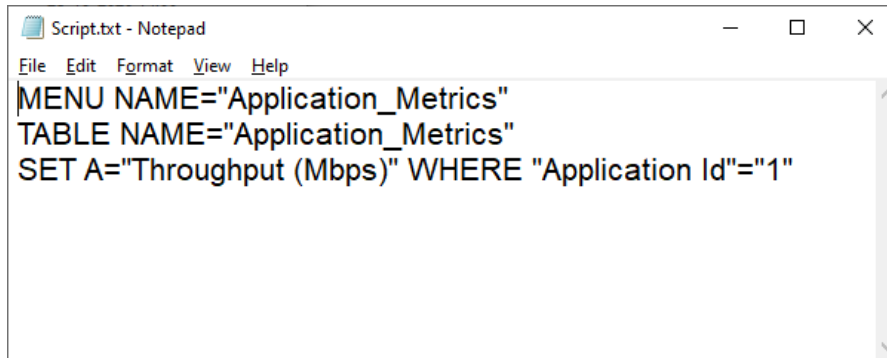
2. **Script.txt:** This file should be updated with the parameter from the output metrics of NetSim that is to be logged at the end of each simulation run for the purpose of analysis. At the end of every simulation, NetSim generates a Metrics.xml file which contain the performance metrics written in a specific format based on which it is loaded in the results dashboard. Each Metric is part of a results table which can be accessed using a menu in the results dashboard. A NetSim Metrics.xml file is shown below:

```

360 <MENU Name="UDP Metrics">...</MENU>
383 <MENU Name="Application Metrics">
384   <TABLE name="Application Metrics">
385     <TH name="Application Id" isShow="true"/>
386     <TH name="Throughput Plot" isShow="true"/>
387     <TH name="Application Name" isShow="true"/>
388     <TH name="Source Id" isShow="false"/>
389     <TH name="Destination Id" isShow="false"/>
390     <TH name="Packet generated" isShow="true"/>
391     <TH name="Packet received" isShow="true"/>
392     <TH name="Payload generated (bytes)" isShow="false"/>
393     <TH name="Payload received (bytes)" isShow="false"/>
394     <TH name="Throughput (Mbps)" isShow="true"/>
395     <TH name="Delay (microsec)" isShow="true"/>
396     <TH name="Jitter (microsec)" isShow="true"/>
397     <TR>
398       <TC Value="1"/>
399       <TC>...</TC>
400     <TC Value="App1_CBR"/>
401     <TC Value="3"/>
402     <TC Value="4"/>
403     <TC Value="25000"/>
404     <TC Value="17946"/>
405     <TC Value="36500000"/>
406     <TC Value="26201160"/>
407     <TC Value="4192.185600"/>
408     <TC Value="7188.952970"/>
409   </TR>
410 </TABLE>

```

For Example, if the application throughput is to be logged for each simulation run then the scrip file can be updated as shown below:



```
Script.txt - Notepad
File Edit Format View Help
MENU NAME="Application_Metrics"
TABLE NAME="Application_Metrics"
SET A="Throughput (Mbps)" WHERE "Application Id"="1"
```

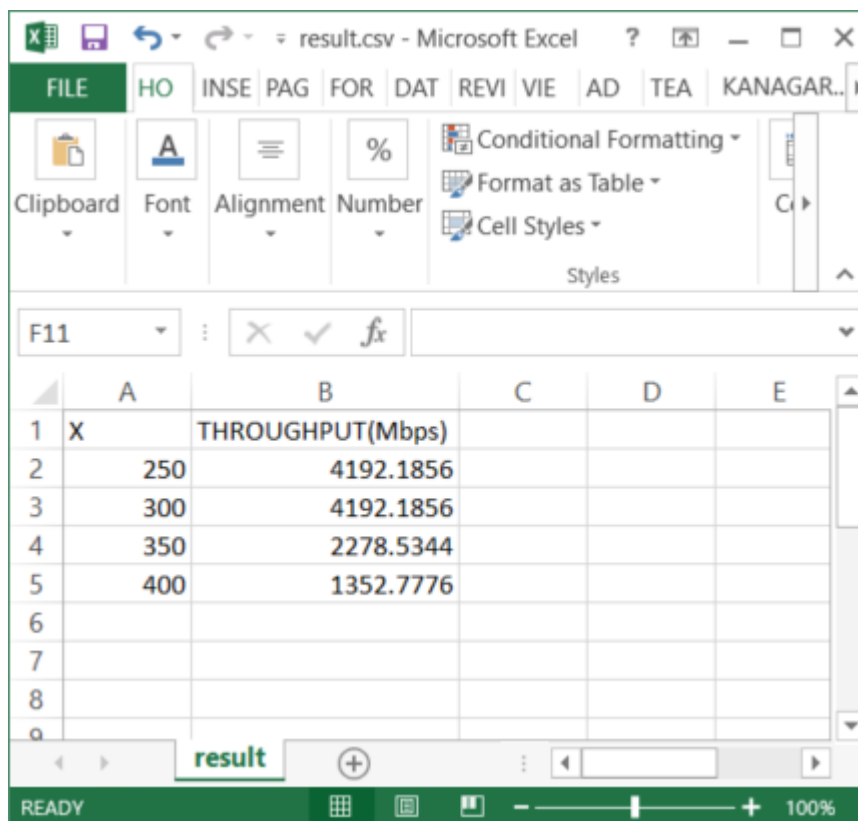
3. **ConfigWriter.exe:** This executable takes one or more command line arguments as input and generated Configuration.netsim file by replacing the arguments in place of the variable parameters specified in the input.xml file.

If there are two variable parameters specified in the input.xml file ({0} and {1}) then two arguments need to be passed while calling ConfigWriter.exe.

4. **MetricsReader.exe:** This executable is responsible for reading the output parameter from the Metrics.xml file generated after each simulation and logging it to the results file.

Users the Script.txt file to determine which parameter to read from the Metrics file.

If multiple parameters are to be read and logged, then the MetricsReader.exe can be called multiple times with Script.txt file having information about the parameter to be read each time.



	A	B	C	D	E
1	X	THROUGHPUT(Mbps)			
2	250	4192.1856			
3	300	4192.1856			
4	350	2278.5344			
5	400	1352.7776			
6					
7					
8					

5. **Supporting DLL's:** Some the supporting files such as ConfigWriter.dll, MetricsReader.dll, NetSimMetricsReader.dll, NetSimXmlReader.dll, etc. which are present in the project folder are used by other executable such as ConfigWriter.exe and MetricsReader.exe for various purposes during a multi-parameter sweep.

6. **runTest** script files: The files runTest.bat, runTest.py are the main script files that can be used to start a multi-parameter sweep process. Both files differ in the programming language used for the script.

runTest.bat uses Windows commands that can be executed by the windows command line interpreter. Batch scripts may get complex as the number of input and output parameters increases.

runTest.py uses python programming language which is less complex and offers more flexibility as the number of input and output parameters increases.

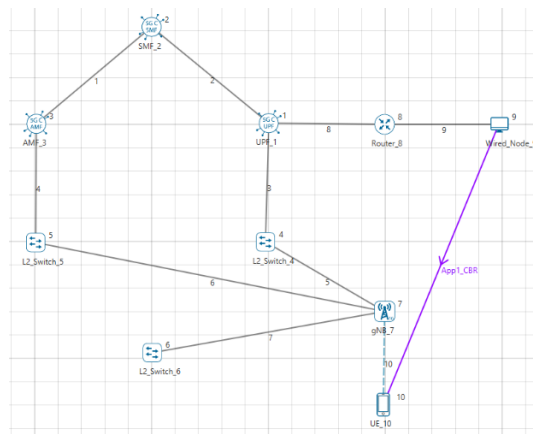
Users can also write the script to run the multi-parameter sweep process in a preferred programming language as per the convenience.

The script runs multiple simulation iterations based on the number of parameters to be varied and the range of values of each parameter.

Running a Multi-Parameter Sweep process:

Example 1: Modifying a single input parameter and logging a single output parameter

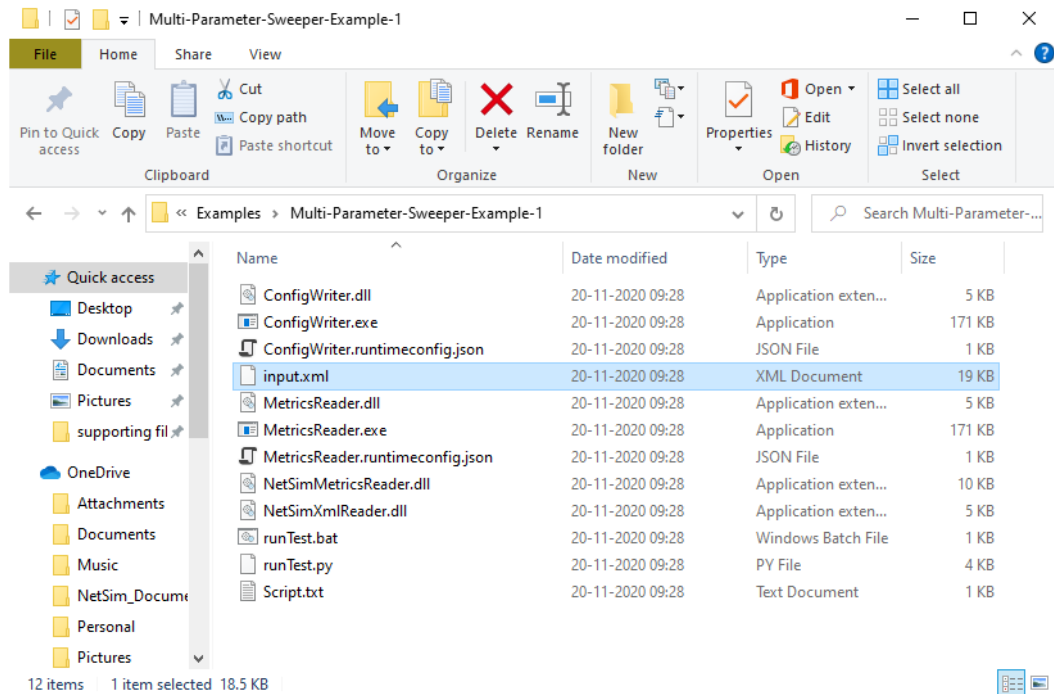
Consider the following network 5G network scenario in NetSim, comprising of a Wired Node, Router, gNB and a UE.



The network configuration has the initial distance between the gNB and UE as 50 meters with the gNB located at (1000,200) and UE located at (1000,250).

Multi-Parameter Sweeper is configured to run simulations for different distance between the gNB and UE by varying the UE Y coordinate value from 250 to 400 in steps of 50 meters.

1. The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml. Refer to the Example 1 directory which is part of the project folder (Multi-Parameter-Sweeper_v13.0\Examples\Multi-Parameter-Sweeper-Example-1)

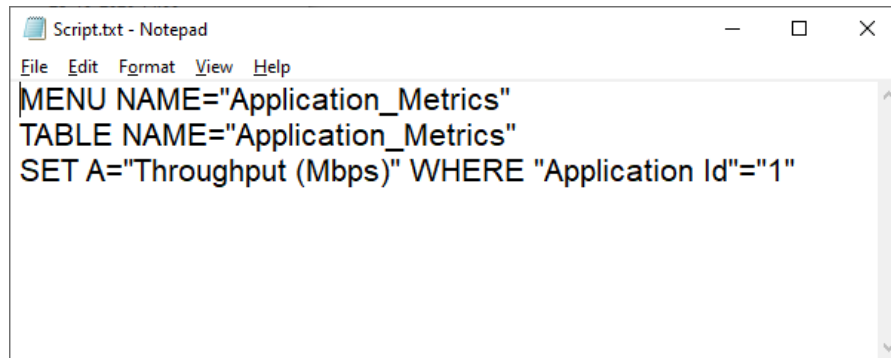


The value of the Y coordinate of UE that is to be modified during each simulation run is updated ("{0}") in the configuration file as shown below:

The screenshot shows a Visual Studio Code editor window with the 'input.xml' file open. The XML content is displayed, and the line containing the Y coordinate is highlighted with a red box. The status bar at the bottom indicates 'Ln: 251 Ch: 1 SPC CRLF'.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TETCOS_NETSIM xmlns:ns0="http://www.w3.org/2001/XMLSchema-instance" ns0:noNamespaceSchemaLocation="Configur...
<EXPERIMENT_INFORMATION>...</EXPERIMENT_INFORMATION>
<GUI_INFORMATION>...</GUI_INFORMATION>
<NETWORK_CONFIGURATION>
  <DEVICE_CONFIGURATION DEVICE_COUNT="4">
    <DEVICE DEFAULT_DEVICE_NAME="gNB" DEVICE_ID="1" DEVICE_IMAGE="gNB.png" DEVICE_NAME="gNB_1" DEVICE_TYPE
    <DEVICE DEFAULT_DEVICE_NAME="EPC" DEVICE_ID="2" DEVICE_IMAGE="EPC.png" DEVICE_NAME="EPC_2" DEVICE_TYPE
    <POS_3D X_OR_LON="804.99" Y_OR_LAT="200.54999999999998" Z="0"/>
    <INTERFACE ID="1" INTERFACE_TYPE="ETHERNET">...</INTERFACE>
    <INTERFACE ID="2" INTERFACE_TYPE="LTE_NR">...</INTERFACE>
    <LAYER_TYPE="APPLICATION_LAYER">...</LAYER>
    <LAYER_TYPE="TRANSPORT_LAYER">...</LAYER>
    <LAYER_TYPE="NETWORK_LAYER">...</LAYER>
  </DEVICE>
  <DEVICE DEFAULT_DEVICE_NAME="Wired_Node" DEVICE_ID="3" DEVICE_IMAGE="WiredNode.png" DEVICE_NAME="Wired
  <DEVICE DEFAULT_DEVICE_NAME="UE" DEVICE_ID="4" DEVICE_IMAGE="UserEquipment.png" DEVICE_NAME="UE_4" DEV
  <POS_3D X_OR_LON="1000" Y_OR_LAT="{0}" Z="0">
    <MOBILITY_MODEL="NO_MOBILITY"/>
  </POS_3D>
  <INTERFACE ID="1" INTERFACE_TYPE="LTE_NR">...</INTERFACE>
  <LAYER_TYPE="APPLICATION_LAYER"/>
  <LAYER_TYPE="TRANSPORT_LAYER">...</LAYER>
</DEVICE>
</DEVICE_CONFIGURATION>
<CONNECTION>...</CONNECTION>
<APPLICATION_CONFIGURATION COUNT="1">...</APPLICATION_CONFIGURATION>
```

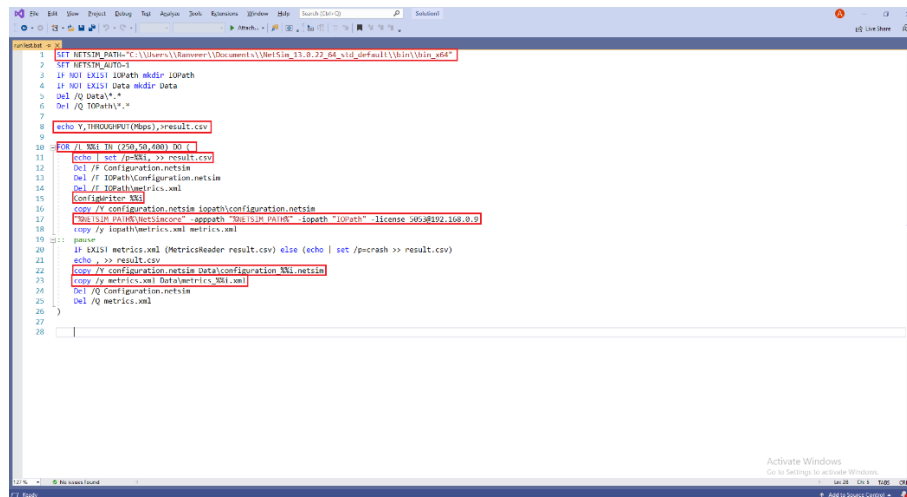
- The Script.txt file is updated with the details of the output parameter to be read from the Metrics.xml file and added to the result csv log file. In this case the Application throughput is to be logged for each simulation run.



```
Script.txt - Notepad
File Edit Format View Help
MENU NAME="Application_Metrics"
TABLE NAME="Application_Metrics"
SET A="Throughput (Mbps)" WHERE "Application Id"="1"
```

- runTest.py/runTest.bat is updated to pass the Y coordinate value during each iteration to generate Configuration file run simulation and update the result csv log.

The runTest.bat batch script modified for running simulations for different values of Y coordinates starting from 250 up to 400 in steps of 50 is shown below:



```
runTest.bat
1 SET NETSIM_PATH="C:\Users\...\Documents\NetSim_32-bit_64-bit_default\bin\Win_64"
2 SET NETSIM_PATH=1
3 IF NOT EXIST %IOPATH% mkdir %IOPATH%
4 IF NOT EXIST %DATA% mkdir %DATA%
5 Del /Q %DATA%\*.*
6 Del /Q %IOPATH%\*.*
7
8 echo Y,THROUGHPUT(Mbps),>result.csv
9
10 FOR /L %Y% IN (250,50,400) DO (
11     echo | set /p=Y %Y% >> result.csv
12     Del /Q Configuration.netsim
13     Del /Q %IOPATH%\Configuration.netsim
14     Del /Q %IOPATH%\Metrics.xml
15     ConfigWriter %Y%
16     copy /Y Configuration.netsim %IOPATH%\Configuration.netsim
17     NetSim /PATH=%NETSIM_PATH% /APPATH="%IOPATH%" /IOPATH="%IOPATH%" /LICENSE="60,60,102,108,0,0"
18     copy /Y %IOPATH%\Metrics.xml %DATA%\metrics.xml
19     pause
20     IF EXIST %DATA%\metrics.xml (MetricsHeader result.csv) else (echo | set /p=crash >> result.csv)
21     echo %Y% >> result.csv
22     copy /Y Configuration.netsim %DATA%\Configuration.netsim
23     copy /Y %DATA%\Data\Metrics.xml %IOPATH%\Metrics.xml
24     Del /Q Configuration.netsim
25     Del /Q %DATA%\metrics.xml
26 )
27
28
```

- NETSIM_PATH variable is set to the path of NetSim 32-bit/64-bit binaries in the install directory or workspace in the system.
- A result.csv file is created and added with headings Y and Throughput(Mbps)
- For loop is set to iteratively run simulations for values starting from 250 to 400 in steps of 50.
- The value of the parameter Y in the current iteration is written to the result log file for analysis.
- The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.
- NetSim simulation is run via CLI mode by passing the apppath, iopath and license server information

- Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.

The runTest.py python script modified for running simulations for different values of Y coordinates starting from 250 up to 400 in steps of 50 is shown below:

```

1 import random
2 import shutil
3 import math
4 import sys
5 import os
6
7 #Set the path of NetSim Binaries to be used for simulation. Either 32 bit or 64 bit
8 NETSIM_PATH="C:\\Users\\Ranveer\\Documents\\NetSim_13.0.22_64_std_default\\bin\\bin_x64"
9
10 #Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
11 os.environ["NETSIM_AUTO"] = '1'
12
13 #Create IOPATH directory to store the input Configuration.netsim file and the simulation output files during each iteration
14 if not os.path.exists("IOPATH"):
15     os.makedirs("IOPATH")
16
17 #Create Data directory to store the Configuration.netsim and the Metrics.xml files associated with each iteration
18 if not os.path.exists("Data"):
19     os.makedirs("Data")
20
21 #Clear the IOPATH folder if it has any files created during previous multi-parameter sweep runs
22 for root, dirs, files in os.walk("IOPATH"):
23     for file in files:
24         os.remove(os.path.join(root, file))
25
26 #Clear the Data folder if it has any files created during previous multi-parameter sweep runs
27 for root, dirs, files in os.walk("Data"):
28     for file in files:
29         os.remove(os.path.join(root, file))
30
31 #Delete result.csv file if it already exists
32 if os.path.isfile("result.csv"):
33     os.remove("result.csv")
34
35 #create a csv file to log the output metrics for analysis
36 csvfile = open("result.csv", "w")
37
38 #Add headings to the CSV file
39 csvFile.write("Y,Throughput(Mbps),")
40 csvFile.close()
41
42 #Iterate based on the number of time simulation needs to be run and the input parameter range

```

- NETSIM_PATH variable is set to the path of NetSim 32-bit/64-bit binaries in the install directory or workspace in the system.
- A result.csv file is created and added with headings Y and Throughput(Mbps)

```

40 csvfile.close()
41
42 #Iterate based on the number of time simulation needs to be run and the input parameter range
43 for i in range(250, 401, 50):
44
45     if os.path.isfile("Configuration.netsim"):
46         os.remove("Configuration.netsim")
47
48     if os.path.isfile("IOPATH\\Configuration.netsim"):
49         os.remove("IOPATH\\Configuration.netsim")
50
51     if os.path.isfile("IOPATH\\Metrics.xml"):
52         os.remove("IOPATH\\Metrics.xml")
53
54     #Call ConfigWriter.exe with arguments as per the number of variable parameters in the input.xml file
55     cmd="ConfigWriter.exe "+str(i)
56     print(cmd)
57     os.system(cmd)
58
59     #Copy the Configuration.netsim file generated by ConfigWriter.exe to IOPATH directory
60     if os.path.isfile("Configuration.netsim"):
61         shutil.copy("Configuration.netsim", "IOPATH\\Configuration.netsim")
62
63     #Run NetSim via CLI mode by passing the apppath iopath and license information to the NetSimCore.exe
64     cmd="NETSIM_PATH"+NETSIM_PATH+"-appath "+NETSIM_PATH+"-iopath IOPATH -license SD538192.168.0.0"
65     os.system(cmd)
66     #print(cmd)
67
68     #Create a copy of the output Metrics.xml file for writing the result log
69     if os.path.isfile("IOPATH\\Metrics.xml"):
70         shutil.copy("IOPATH\\Metrics.xml", "Metrics.xml")
71
72     #Write the value of the variable parameter in the current iteration to the result log
73     csvfile = open("result.csv", "a")
74     if os.path.isfile("Metrics.xml"):
75         csvFile.write("in"+str(i)+",")
76         csvfile.close()
77
78     #Call the MetricsReader.exe passing the name of the output log file for updating the log based on script.txt
79     os.system("MetricsReader.exe result.csv")
80
81 else:
82     #Update the output Metric as crash if Metrics.xml file is missing

```

- For loop is set to iteratively run simulations for values starting from 250 to 400 in steps of 50.
- The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.

- NetSim simulation is run via CLI mode by passing the apppath, iopath and license server information.
- The value of the parameter Y in the current iteration is written to the result log file for analysis.

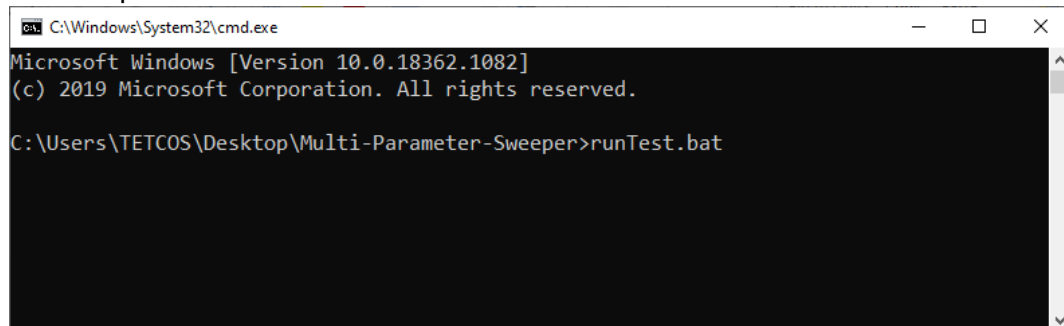
```

79
80 else:
81     #Update the output Metric as crash if Metrics.xml file is missing
82     csvfile.write('\n'+str(i)+'+'crash'+',')
83     csvfile.close()
84
85 #Create a copy of the Configuration.netsim file appending the value of the variable parameters used in the current iteration
86 if(os.path.isfile("Configuration.netsim")):
87     shutil.copy("Configuration.netsim", "Data\configuration_"+str(i)+".netsim")
88
89 #Create a copy of the Metrics.xml file appending the value of the variable parameters used in the current iteration
90 if(os.path.isfile("Metrics.xml")):
91     shutil.copy("Metrics.xml", "Data\metrics_"+str(i)+".xml")
92
93 #Delete Configuration.netsim file created during the last iteration
94 if(os.path.isfile("Configuration.netsim")):
95     os.remove("Configuration.netsim")
96
97 #Delete Metrics.xml file created during the last iteration
98 if(os.path.isfile("Metrics.xml")):
99     os.remove("Metrics.xml")
100

```

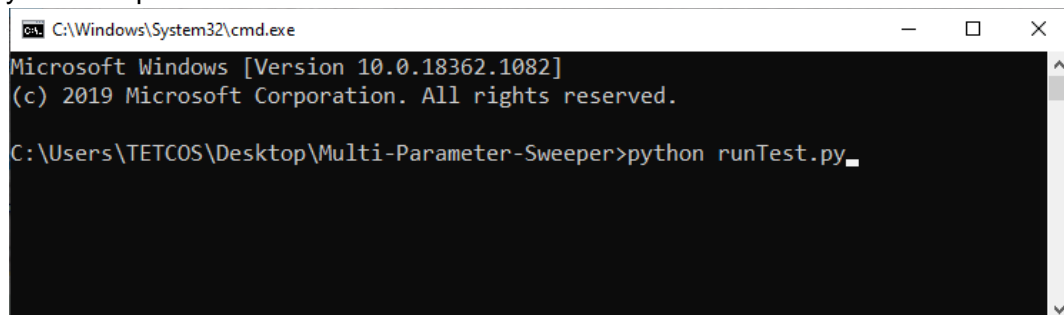
- Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.
4. Multi-Parameter Sweeping process is started by opening command prompt in the directory of the Multi-Parameter-Sweeping project and starting the batch script or the python script as shown below:

Batch Script:



A screenshot of a Windows command prompt window titled "C:\Windows\System32\cmd.exe". The window shows the following text: "Microsoft Windows [Version 10.0.18362.1082] (c) 2019 Microsoft Corporation. All rights reserved. C:\Users\TETCOS\Desktop\Multi-Parameter-Sweeper>runTest.bat". The command prompt is currently at the end of the command line.

Python Script:



A screenshot of a Windows command prompt window titled "C:\Windows\System32\cmd.exe". The window shows the following text: "Microsoft Windows [Version 10.0.18362.1082] (c) 2019 Microsoft Corporation. All rights reserved. C:\Users\TETCOS\Desktop\Multi-Parameter-Sweeper>python runTest.py". The command prompt is currently at the end of the command line.

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of UE.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

Name	Date modified	Type
Data	28-10-2020 19:20	File folder
IOPath	28-10-2020 19:20	File folder
ConfigWriter.dll	22-10-2020 00:45	Application extension
ConfigWriter.exe	22-10-2020 00:45	Application
ConfigWriter.runtimeconfig.json	22-10-2020 00:45	JSON File
input.xml	28-10-2020 13:17	XML Document
MetricsReader.dll	22-10-2020 00:45	Application extension
MetricsReader.exe	22-10-2020 00:45	Application
MetricsReader.runtimeconfig.json	22-10-2020 00:45	JSON File
NetSimMetricsReader.dll	22-10-2020 00:45	Application extension
NetSimXmlReader.dll	22-10-2020 00:45	Application extension
result.csv	28-10-2020 19:20	Microsoft Excel Comma...
runTest.bat	28-10-2020 19:11	Windows Batch File
runTest.py	28-10-2020 19:12	Python File
Script.txt	24-10-2020 12:15	Text Document

- **Data:** The Data directory contains the Configuration.netsim and the Metrics.xml files associated with each simulation run, renamed including the value of the parameter in the file name.

Name	Date modified	Type
configuration_250.netsim	28-10-2020 19:18	NetSim_File
configuration_300.netsim	28-10-2020 19:18	NetSim_File
configuration_350.netsim	28-10-2020 19:19	NetSim_File
configuration_400.netsim	28-10-2020 19:20	NetSim_File
metrics_250.xml	28-10-2020 19:18	XML Document
metrics_300.xml	28-10-2020 19:18	XML Document
metrics_350.xml	28-10-2020 19:19	XML Document
metrics_400.xml	28-10-2020 19:20	XML Document

- **IOPath:** Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
- **Result.csv:** This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.

The screenshot shows a Microsoft Excel window titled 'result.csv - Microsoft Excel'. The ribbon is set to 'HOME' with tabs for FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, VIEW, ADD-INS, and KANAGAR... The ribbon includes groups for Clipboard, Font, Alignment, Number, and Styles. The Styles group contains 'Conditional Formatting', 'Format as Table', and 'Cell Styles'. The formula bar shows 'I14' and a function 'f_x'. The spreadsheet has columns A through E and rows 1 through 8. The data is as follows:

	A	B	C	D	E
1	Y	THROUGHPUT(Mbps)			
2	250	4192.1856			
3	300	4192.1856			
4	350	2278.5344			
5	400	1352.7776			
6					
7					
8					

The status bar at the bottom shows 'READY' and a zoom level of 100%.

Varying multiple network parameters:

In order to vary multiple network parameters during the multi-parameter sweep process each parameter in the input.xml file can be modified as $\{0\}, \{1\}, \{2\}, \{3\}, \dots, \{n\}$ respectively.

Logging multiple output parameters:

Each output parameter that is to be logged should be part of the Script.txt file. However, the Script.txt file should contain only the details of one output parameter during the call to MetricsReader.exe.

To log multiple parameters, MetricsReader.exe can be called multiple times with Script.txt file having information about different parameter during each call.

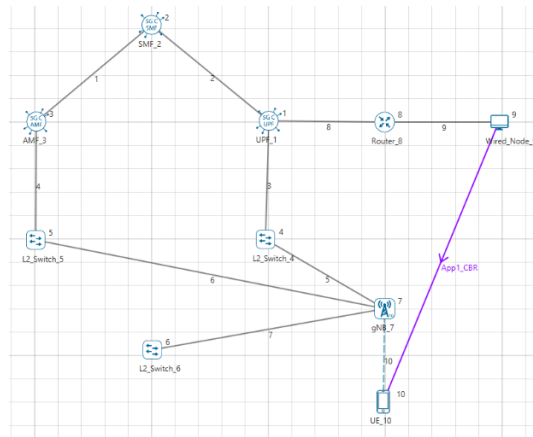
For Example, there can be two Script files as shown below:

Name	Date modified	Type
Data	28-10-2020 19:20	File folder
IOPath	28-10-2020 19:20	File folder
ConfigWriter.dll	22-10-2020 00:45	Application extension
ConfigWriter.exe	22-10-2020 00:45	Application
ConfigWriter.runtimeconfig.json	22-10-2020 00:45	JSON File
input.xml	28-10-2020 13:17	XML Document
MetricsReader.dll	22-10-2020 00:45	Application extension
MetricsReader.exe	22-10-2020 00:45	Application
MetricsReader.runtimeconfig.json	22-10-2020 00:45	JSON File
NetSimMetricsReader.dll	22-10-2020 00:45	Application extension
NetSimXmlReader.dll	22-10-2020 00:45	Application extension
result.csv	28-10-2020 19:20	Microsoft Excel Comma...
runTest.bat	28-10-2020 18:56	Windows Batch File
runTest.py	28-10-2020 19:12	Python File
Script1.txt	24-10-2020 12:15	Text Document
Script2.txt	24-10-2020 12:15	Text Document

During each call to MetricsReader each of the Script files (Script1.txt and Script2.txt) can be renamed to Script.txt and renamed back.

Example 2: Modifying multiple input parameters and logging multiple output parameter

Consider the following network 5G network scenario in NetSim, comprising of a Wired Node, Router, gNB and a UE.



Properties configured in the LTE_NR interface of the gNB is shown in the table below:

Interface(5G_RAN) Properties	
Tx_Power(dBM)	40
Tx_Antenna_Count	8
Rx_Antenna_Count	4
CA_Type	Single Band
CA_Configuration	n78
CA_Count	1
MU	0
Channel Bandwidth (MHz)	10
PRB Count	52
MCS Table	QAM64
CQI Table	Table 1
X_Overhead	XOH0
DL_UL_Ratio	4:1
Outdoor Scenario	Rural Macro
LOS Mode	Standard
Wireless Link Properties	
Channel Characteristics	No_Pathloss
Wired Link Properties	
Link Speed (Mbps)	10000
BER	0
Propagation Delay (μs)	0
Application Properties	
Packet Size (Byte)	1460
Inter Arrival Time (μs)	166
Generation Rate (Mbps)	100

Transport Control	UDP
Start Time (s)	1
QoS	BE
Simulation Parameters	
Simulation Time (s)	1.1

Traffic is generated at a rate of 70 Mbps and upon running simulation, the throughput achieved is 59.95 Mbps.

We now find the max throughput for each possible bandwidth, Tx Antenna count and Rx Antenna count combination varying the generation rate based accordingly.

Two more parameters to be taken care include, the PRB Count and Guard Band(KHz) which vary with respect to the bandwidth.

Input Variables	Value Range
Channel Bandwidth (MHz)	10,15,20,25,30,40,50
Tx_Antenna_Count	1,2,4,8,16,32,64,128
Rx_Antenna_Count	1,2,4,8,16
PRB Count	52,79,106,133,160,216,270
Guard Band (KHz)	312.5,382.5,452.5,522.5,592.5,552.5,692.5
Reference Inter Arrival Time (Micro Seconds)	166
Reference Bandwidth	10
Reference DL MIMO Layer Count	2

Inter Arrival Time for each case is calculated based on the Reference IAT Bandwidth and DL MIMO Layer Count as shown below:

$$Inter\ Arrival\ Time\ (Micro\ Seconds) = \frac{Ref\ IAT}{\left(\frac{Curr\ BW}{Ref\ BW}\right) * \left(\frac{Curr\ DL\ MIMO\ Count}{Ref\ DL\ MIMO\ Count}\right)}$$

For E.g. In case of Bandwidth of 20 MHz and DL MIMO Count of 4 inter arrival time is

$$Inter\ Arrival\ Time\ (Micro\ Seconds) = \frac{166}{\left(\frac{20}{10}\right) * \left(\frac{4}{2}\right)} = 41.5\ Mbps$$

1. The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.
2. Refer to the Example 2 directory which is part of the project folder (Multi-Parameter-Sweeper_v13.0\Examples\Multi-Parameter-Sweeper-Example-2)

- The python script runTest.py is modified to run simulation for all possible combinations of Bandwidth and Tx Antenna Count and Rx Antenna Count with the respective values of Guard Band, PRB Count and the IAT that is calculated.

```

16 for root, dirs, files in os.walk("Data"):
17     for file in files:
18         os.remove(os.path.join(root, file))
19
20 for root, dirs, files in os.walk("IOPath"):
21     for file in files:
22         os.remove(os.path.join(root, file))
23
24 bandwidth=[10,15,20,25,30,40,50]
25 Tx_Antenna_count=[8,16,32,64,128]
26 Rx_Antenna_count = [2,4,8,16]
27 prb_count=[52,79,106,133,160,216,270]
28 guard_band=[312.5,382.5,452.5,522.5,592.5,662.5]
29
30 ref_iat=166
31
32 if os.path.isfile("result.csv"):
33     os.remove("result.csv")
34
35 csvfile = open("result.csv", 'w')
36 csvfile.write('CHANNELBANDWIDTH_MHz,Tx_ANTENNA_COUNT,Rx_ANTENNA_COUNT,INTER_ARRIVAL_TIME(micro sec),THROUGHPUT(Mbps),Data Packets transmitted,')
37 csvfile.close()
38
39 k=0
40 for i in bandwidth:
41     k+=1
42     for j in Tx_Antenna_count:
43         for z in Rx_Antenna_count:
44             iat=ref_iat/((i/10)*(j/2))
45
46             if os.path.isfile("Configuration.netsim"):
47                 os.remove("Configuration.netsim")
48
49             if os.path.isfile("IOPath\\Configuration.netsim"):
50                 os.remove("IOPath\\Configuration.netsim")
51
52             if os.path.isfile("IOPath\\Metrics.xml"):
53                 os.remove("IOPath\\Metrics.xml")
54
55             cmd="ConfigWriter.exe "+str(i)+" "+str(j)+" "+str(z)+" "+str(iat)+" "+str(prb_count[k-1])\
56                 +" "+str(guard_band[k-1])
57             os.system(cmd)

```

- Multiple parameters are read from the Metrics.xml file and logged in the results.csv file along with the input parameters such as CHANNELBANDWIDTH_MHz, TX_ANTENNA_COUNT, RX_ANTENNA_COUNT, INTER_ARRIVAL_TIME (micro sec)

Output Parameters
Throughput(Mbps)
Data Packets transmitted

- Two script text files namely Script1.txt and Script2.txt are created with information to read each of the parameters from the Metrics.xml file.

Script1.txt

```

Script1.txt - Notepad
File Edit Format View Help
MENU NAME="Application_Metrics"
TABLE NAME="Application_Metrics"
SET A="Throughput (Mbps)" WHERE "Application Id"="1"

```

Script2.txt

```

Script2.txt - Notepad
File Edit Format View Help
MENU NAME="Link_Metrics"
TABLE NAME="Link_Metrics"
SET A="Packet_transmitted\\Data" WHERE "Link_id"="All"

```

- In the python script runTest.py, MetricsReader is called twice, once to log each parameter by renaming the script files to Script.txt and back to its original name and separating the entries with a comma (","). The input parameters that were varied during each simulation run are also logged in the results.csv file.

```

52 if(os.path.isfile("IOPath\Metrics.xml")):
53     os.remove("IOPath\Metrics.xml")
54
55 cmd="Configuration.exe "+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])\
56     +" "+str(bandwidth[i])
57 print(cmd)
58 os.system(cmd)
59
60 if(os.path.isfile("Configuration\netsim")):
61     shutil.copy("Configuration\netsim","IOPath\Configuration\netsim")
62
63 cmd="NETSIM_PATH\\Vetsimcore.exe -appath "%NETSIM_PATH%" -iopath IOPath -license %GSAPRQ.160.0.0"
64 os.system(cmd)
65 print(cmd)
66
67 if(os.path.isfile("IOPath\Metrics.xml")):
68     shutil.copy("IOPath\Metrics.xml","Metrics.xml")
69
70 if(os.path.isfile("Metrics.xml")):
71     csvfile = open("result.csv","a")
72     csvfile.write("\n"+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+"\n")
73     os.rename("Script1.txt","Script.txt")
74     csvfile.close()
75     os.system("MetricsReader.exe result.csv")
76     csvfile = open("result.csv","a")
77     csvfile.write(",")
78     csvfile.close()
79     os.rename("Script.txt","Script1.txt")
80     os.rename("Script2.txt","Script.txt")
81     os.system("MetricsReader.exe result.csv")
82     os.rename("Script.txt","Script2.txt")
83
84 else:
85     csvfile.write("\n"+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+"\n")
86     csvfile.close()
87
88 if(os.path.isfile("Configuration\netsim")):
89     shutil.copy("Configuration\netsim","Data\Configuration_"+str(i)\
90         +" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+".netsim")
91
92 if(os.path.isfile("Metrics.xml")):
93     shutil.copy("Metrics.xml","Data\Metrics_"+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+".xml")
94
95 if(os.path.isfile("Configuration\netsim")):
96     os.remove("Configuration\netsim")
97
98 if(os.path.isfile("Metrics.xml")):
99     os.remove("Metrics.xml")
100
101

```

- The Input Configuration.netsim and the output Metrics.xml files associated with each simulation run is renamed including the bandwidth and DL MIMO count values that were used during each simulation run.

```

71 csvfile = open("result.csv","a")
72 csvfile.write("\n"+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+"\n")
73 os.rename("Script1.txt","Script.txt")
74 csvfile.close()
75 os.system("MetricsReader.exe result.csv")
76 csvfile = open("result.csv","a")
77 csvfile.write(",")
78 csvfile.close()
79 os.rename("Script.txt","Script1.txt")
80 os.rename("Script2.txt","Script.txt")
81 os.system("MetricsReader.exe result.csv")
82 os.rename("Script.txt","Script2.txt")
83
84 else:
85     csvfile.write("\n"+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+"\n")
86     csvfile.close()
87
88 if(os.path.isfile("Configuration\netsim")):
89     shutil.copy("Configuration\netsim","Data\Configuration_"+str(i)\
90         +" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+".netsim")
91
92 if(os.path.isfile("Metrics.xml")):
93     shutil.copy("Metrics.xml","Data\Metrics_"+str(i)+" "+str(j)+" "+str(iat)+" "+str(prb_count[i-1])+".xml")
94
95 if(os.path.isfile("Configuration\netsim")):
96     os.remove("Configuration\netsim")
97
98 if(os.path.isfile("Metrics.xml")):
99     os.remove("Metrics.xml")
100
101

```

- Multi-Parameter Sweeping process is started by opening command prompt in the directory of the Multi-Parameter-Sweeper project and starting the python script as shown below:

Python Script:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\TETCOS\Desktop\Multi-Parameter-Sweeper>python runTest.py

```


This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different combinations of input parameters.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

Name	Date modified	Type
Data	19-11-2020 23:04	File folder
IOPath	19-11-2020 23:04	File folder
ConfigWriter.dll	22-10-2020 00:45	Application extension
ConfigWriter.exe	22-10-2020 00:45	Application
ConfigWriter.runtimeconfig.json	22-10-2020 00:45	JSON File
input.xml	19-11-2020 22:46	XML Document
MetricsReader.dll	22-10-2020 00:45	Application extension
MetricsReader.exe	22-10-2020 00:45	Application
MetricsReader.runtimeconfig.json	22-10-2020 00:45	JSON File
NetSimMetricsReader.dll	22-10-2020 00:45	Application extension
NetSimXmlReader.dll	22-10-2020 00:45	Application extension
result.csv	19-11-2020 23:04	Microsoft Excel Comma...
runTest.py	19-11-2020 22:56	PY File
Script1.txt	24-10-2020 12:15	Text Document
Script2.txt	19-11-2020 17:42	Text Document

- **Data:** The Data directory contains the Configuration.netsim and the Metrics.xml files associated with each simulation run, renamed including the value of the parameter in the file name.

Name	Date modified	Type	Size
configuration_10_2.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_10_4.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_10_8.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_15_2.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_15_4.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_15_8.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_20_2.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_20_4.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_20_8.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_25_2.netsim	19-11-2020 23:03	NetSim_File	15 KB
configuration_25_4.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_25_8.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_30_2.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_30_4.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_30_8.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_40_2.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_40_4.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_40_8.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_50_2.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_50_4.netsim	19-11-2020 23:04	NetSim_File	15 KB
configuration_50_8.netsim	19-11-2020 23:04	NetSim_File	15 KB
metrics_10_2.xml	19-11-2020 23:03	XML Document	11 KB
metrics_10_4.xml	19-11-2020 23:03	XML Document	11 KB

- **IOPath:** Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
- **Result.csv:** This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.

AutoSave result.csv					
File Home Insert Page Layout Formulas Data Review View Help					
<div> <div>Clipboard</div> <div>Font</div> <div>Alignment</div> <div>Number</div> <div>Styles</div> <div>Cells</div> <div>Editing</div> <div>Ideas</div> </div>					
N26					
	A	B	C	D	E
1	CHANNELBANDWIDTH_MHz	DOWNLINK_MIMO_LAYER_COUNT	INTER_ARRIVAL_TIME(micro sec)	THROUGHPUT(Mbps)	Data Packets transmitted
2	10	2	166	59.8016	1790
3	10	4	83	120.4208	3518
4	10	8	41.5	238.0384	6932
5	15	2	110.6666667	90.9872	2663
6	15	4	55.33333333	183.1424	5260
7	15	8	27.66666667	362.1968	10405
8	20	2	83	123.4576	3542
9	20	4	41.5	248.5504	7023
10	20	8	20.75	491.4944	13922
11	25	2	66.4	154.6432	4403
12	25	4	33.2	311.272	8765
13	25	8	16.6	615.6528	17393
14	30	2	55.33333333	185.8288	5282
15	30	4	27.66666667	373.9936	10508
16	30	8	13.83333333	739.8112	20866
17	40	2	41.5	250.7696	7042
18	40	4	20.75	504.8096	14038
19	40	8	10.375	998.5232	27901
20	50	2	33.2	314.4256	8780
21	50	4	16.6	632.9392	17542
22	50	8	8.3	1251.9792	34889